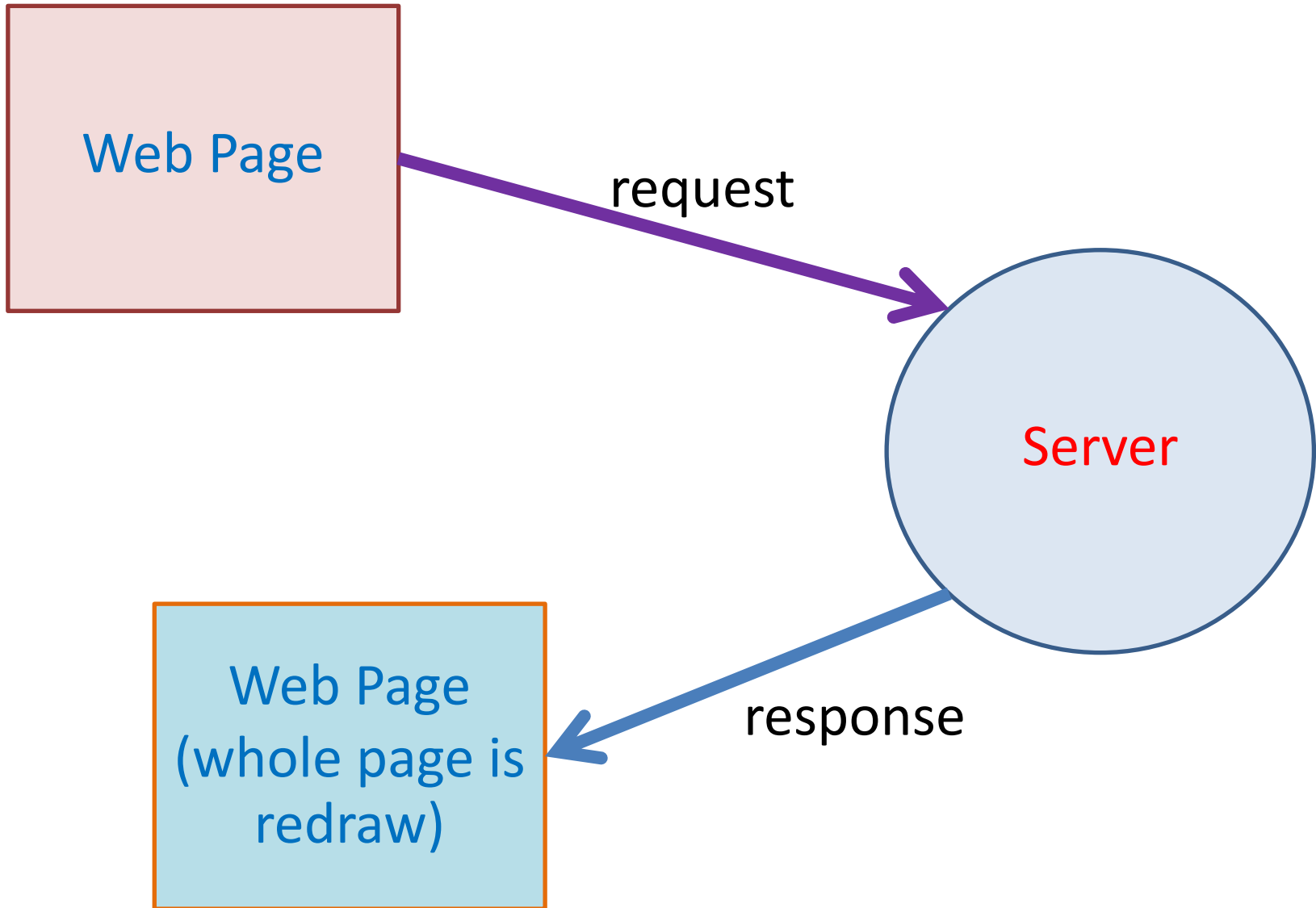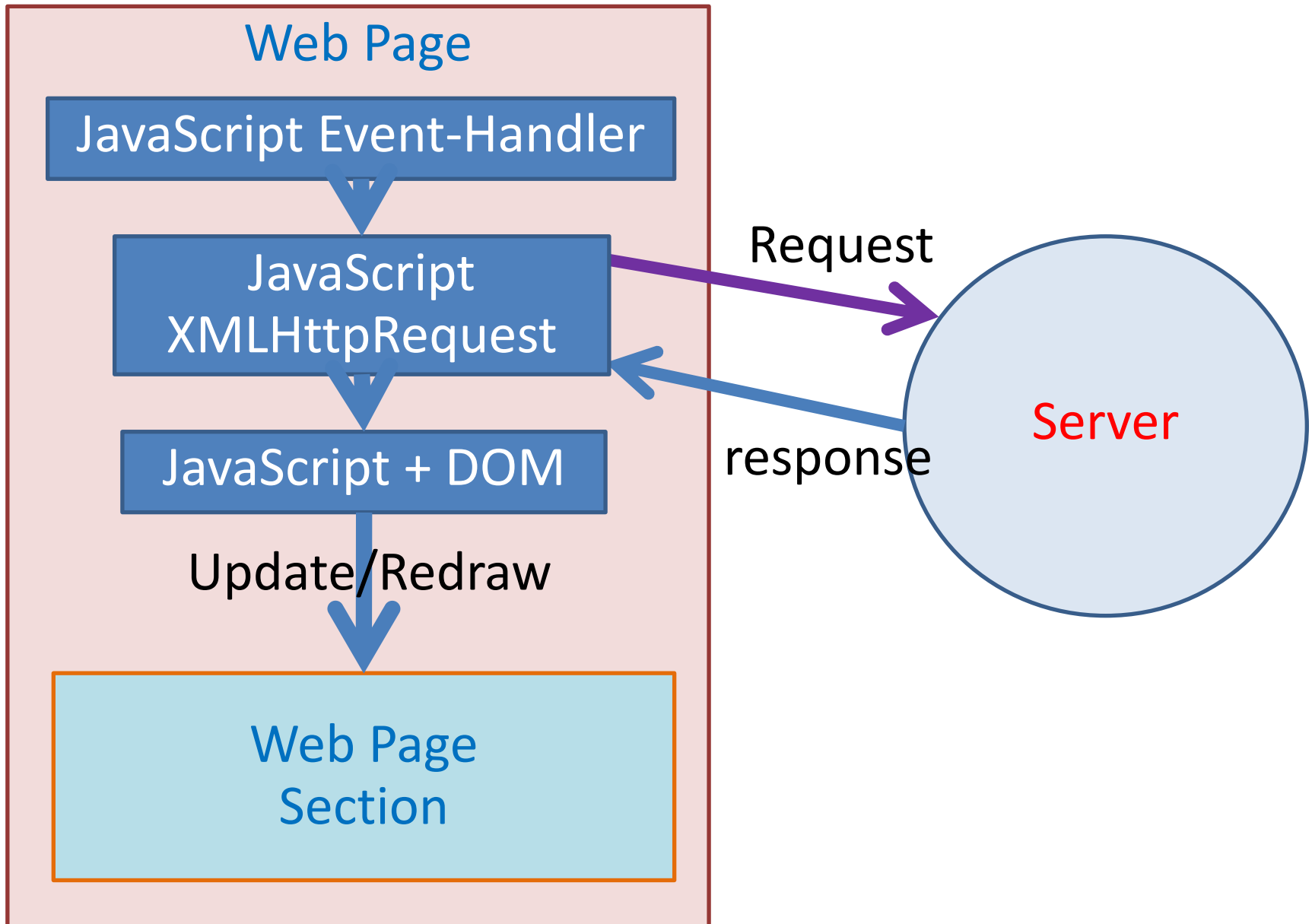# AJAX

Asynchronous JavaScript and XML

- A technology based on:
  - JavaScript
  - DOM
  - Server side script

- Concept: **Client-Driven Communication, Server-Side Processing**

Classical case

Ajax case

Web Page

JavaScript Event-Handler

JavaScript XMLHttpRequest

Request

response

JavaScript + DOM

Update/Redraw

Web Page Section

Server

Client-Side Communication

- Achieved by
  - XMLHttpRequest Object
    - XMLHttpRequest Object Methods

Method	Description
abort()	Cancels the current request
getAllResponseHeaders()	Returns all HTTP headers as a String type variable
getResponseHeader()	Returns the value of the HTTP header specified in the method
open()	Specifies the different attributes necessary to make a connection to the server; allows you to make selections such as GET or POST (more on that later), whether to connect asynchronously, and which URL to connect to
setRequestHeader()	Adds a label/value pair to the header when sent
send()	Sends the current request

- XMLHttpRequest Object Properties

Property	Description
onreadystatechange	Used as an event handler for events that trigger upon state changes
readyState	Contains the current state of the object (0: uninitialized, 1: loading, 2: loaded, 3: interactive, 4: complete)
responseText	Returns the response in string format
responseXML	Returns the response in proper XML format
status	Returns the status of the request in numerical format (regular page errors are returned, such as the number 404, which refers to a not found error)
statusText	Returns the status of the request, but in string format (e.g., a 404 error would return the string Not Found)

- Cross Browser Usage
  - IE and non-IE

```
var xmlhttp = false;

try {
      xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
      alert ("You are using Microsoft Internet Explorer.");
} catch (e) {
      xmlhttp = false;
}

if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
      xmlhttp = new XMLHttpRequest();
      alert ("You are not using Microsoft Internet Explorer");
}
```

```javascript
var xmlhttp;

if (window.ActiveXObject){
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
} else {
    xmlhttp = new XMLHttpRequest();
}
```

- Sending a request to Server

```
function makerequest(serverPage, objID) {
    var obj = document.getElementById(objID);
    xmlhttp.open("GET", serverPage);
    xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
        {           obj.innerHTML = xmlhttp.responseText;
        }
    }
    xmlhttp.send(null);
}
```

The .innerHTML is useful for returning or replacing the content of HTML elements

.innerHTML is not part of W3C DOM but it is supported by all major browser

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Sample 2_1</title>
    <meta http-equiv="Content-Type"
      content="text/html; charset=iso-8859-1" />
    <script type="text/javascript">
    <!--
        var xmlhttp = false;
        try {
            xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
            alert ("You are using MS IE.");
        } catch (e) {
            xmlhttp = false;
        }

        if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
            xmlhttp = new XMLHttpRequest();
            alert ("You are not using MS IE");
        }
```
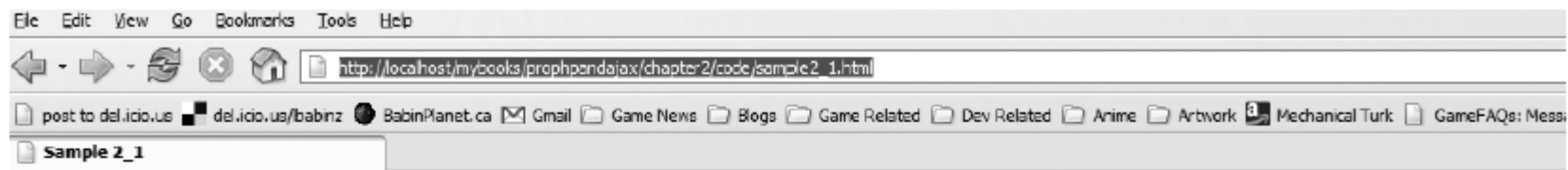
```
function makerequest(serverPage, objID) {
     var obj = document.getElementById(objID);
     xmlhttp.open("GET", serverPage);
     xmlhttp.onreadystatechange = function() {
          if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
          {
                    obj.innerHTML = xmlhttp.responseText;
          }
     }
     xmlhttp.send(null);
       }
    //-->
    </script>
  </head>
```

```html
<body onload="makerequest ('content1.html','hw')">
    <div align="center">
      <h1>My Webpage</h1>
      <a href="content1.html" onclick=
            "makerequest('content1.html','hw');
            return false;"> Page 1</a> |
      <a href="content2.html" onclick=
            "makerequest('content2.html','hw');
            return false;">Page 2</a> |
      <a href="content3.html" onclick=
            "makerequest('content3.html','hw');
            return false;">Page 3</a> |
      <a href="content4.html" onclick=
            "makerequest('content4.html','hw');
            return false;">Page 4</a>
      <div id="hw"></div>
    </div>
  </body>
</html>
```

File   Edit   View   Go   Bookmarks   Tools   Help

http://localhost/mybooks/prophpandajax/chapter2/code/sample2_1.html

post to del.icio.us   del.icio.us/lbabinz   BabinPlanet.ca   Gmail   Game News   Blogs   Game Related   Dev Related   Anime   Artwork   Mechanical Turk   GameFAQs: Messa

Sample 2_1

My Webpage

Page 1 | Page 2 | Page 3 | Page 4

Page 3

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ASP.Net Core MVC & Ajax Example

Controller

```
0 references
public IActionResult Index()
{
    return View();
}

0 references
public IActionResult Message01()
{
    string msg = "Hello,This is a test Ajax message";
    return Content(msg, "text/html", Encoding.UTF8);
}

    2
0 references
public IActionResult Message02()
{
    string msg = "<ul><li>topic01</li><li>topic02</li><li>topic03</li><li>topic04</li><li>topic05</li></ul>";
    return Content(msg, "text/plain", Encoding.UTF8);
}
```

ASP.Net Core MVC & Ajax Example

```html
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Ajax testing</title>
    <script>
        var xmlhttp = new XMLHttpRequest();
        ...
        function MakeReq(serverpage, objID) {
            let obj = document.getElementById(objID);
            xmlhttp.open("GET", serverpage);
            xmlhttp.onreadystatechange = function () {
                if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
                    obj.innerHTML = xmlhttp.responseText;
                }
            }
            xmlhttp.send(null);
        }
    </script>
</head>
<body>
    <div>
        <h2>Part I</h2>
        click here ==>
        <a href="" onclick="MakeReq('https://localhost:5001/Ajax/Message01', 'p01'); return false;">Hello</a>
        <div id="p01"></div>
    </div>
    <div>
        <h2>Part II</h2>
        click here ==>
        <a href="" onclick="MakeReq('https://localhost:5001/Ajax/Message02', 'p02'); return false;">Test</a>
        <div id="p02"></div>
    </div>
</body>
</html>
```